EXAM 2

Solutions

Question 1 (5+5=10 points)

2

(A) Based on the description provided, identify an appropriate data structure:

- Items stored in this structure are located in contiguous locations in the computer's memory.

  Answer: **array or ArrayList**

- The number of items stored in this contiguous storage, sequential structure can be determined by the **length** attribute.

  Answer: **array**

- The number of items stored in this contiguous storage, sequential structure can be determined by the predefined **size()** function.

  Answer: **ArrayList**

- Items added to this structure can only be removed in the last-in-first-out order.

  Answer: **Stack**

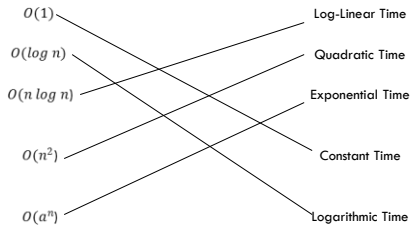
- Items added to this structure can only be removed in the first-in-first-out order.

  Answer: **Queue**

Question 1 (5+5=10 points)

3

- **(B) Match (by drawing lines) the items on the left with complexity characterizations on the right as appropriate:**



Question 2 (5+5+5=15 points)

4

A variable L which is an instance of a class that implements the List<E> interface is used to store items using the following:

```
for (int i=1; i <= 5; i++) {
   L.add(2*i);
}
```

In the pictures below, clearly identify all essential components of the data structure used.

(A) Suppose L is defined as shown below:

```
ArrayList<Integer> L = new ArrayList<Integer>();
```

Draw a picture of the resulting structure showing the items added above:

Question 2 (5+5+5=15 points)

5

A variable L which is an instance of a class that implements the List<E> interface is used to store items using the following:

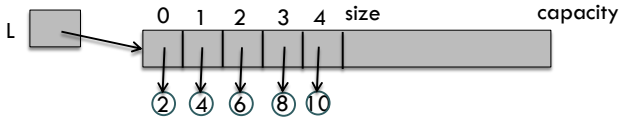
```
for (int i=1; i <= 5; i++) {
   L.add(2*i);
}
```

In the pictures below, clearly identify all essential components of the data structure used.

(A) Suppose L is defined as shown below:

```
ArrayList<Integer> L = new ArrayList<Integer>();
```

Draw a picture of the resulting structure showing the items added above:



Question 2 (5+5+5=15 points)

6

A variable L which is an instance of a class that implements the List<E> interface is used to store items using the following:

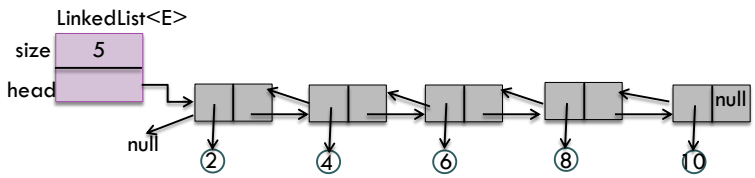
```
for (int i=1; i <= 5; i++) {
   L.add(2*i);
}
```

In the pictures below, clearly identify all essential components of the data structure used.

(B) Suppose L is defined as shown below:

```
LinkedList<Integer> L = new LinkedList<Integer>();
```

Draw a picture of the resulting structure showing the items added above:



Question 2 (5+5+5=15 points)

7

A variable L which is an instance of a class that implements the List<E> interface is used to store items using the following:

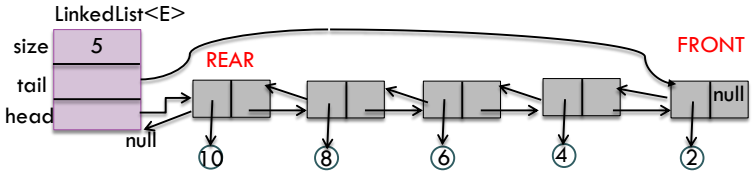
```
for (int i=1; i <= 5; i++) {
   L.add(2*i);
}
```

In the pictures below, clearly identify all essential components of the data structure used.

(C) Suppose L is defined as shown below:

```
Queue<Integer> L = new LinkedList<Integer>();
```

Draw a <u>picture of the resulting structure</u> (it is a linked list, with head and tail stored, front of queue is at tail and rear is at head) showing the items added above):



Question 3 (10+5 = 15 points)

8

Given the class Place as defined in your assignments. The following array contains all the **Place** objects input from a file:

```
ArrayList<Place> places = new ArrayList<Place>();
…
// place is now filled up with place objects.
```

(A) Write the <u>complete definition of a function called searchAll(…)</u> that, given a town and a state, searches through the places list as defined above (and in your assignments) to return a list of all the zip codes used for that town and state. For example, for

```
town = "Cambridge";
state = "MA";
```

it will return a list containing the following zip codes (all of which belong to Cambridge, MA):

02138
02139
12140
02141
02142

Note: Only the function specified needs to be defined.

Question 3 (10+5 = 15 points)

9

Given the class Place as defined in your assignments. The following array contains all the **Place** objects input from a file:

```
ArrayList<Place> places = new ArrayList<Place>();
…
// place is now filled up with place objects.
```

(A) Write the complete definition of a function called searchAll(…) that, given a town and a state, searches through the places list as defined above (and in your assignments) to return a list of all the zip codes used for that town and state. For example, for

```
town = "Cambridge";
state = "MA";
```

it will return a list containing the following zip codes (all of which belong to Cambridge, MA):

```
02138
02139
12140
02141
02142
```

```
public static ArrayList<String> searchAll(ArrayList<Place> p, String town, String zip) {
…
} // searchAll()
```

Note: Only the function specified needs to be defined.

Question 3 (10+5 = 15 points)

10

```
public static ArrayList<String> searchAll(ArrayList<Place> p, String town, String zip) {
        ArrayList<String> result = new ArrayList<String>();
        for (Place pl : p) {
                if (town.equalsIgnoreCase(p.getTown())
                  && lzip.equalsIgnoreCase(p.getState()) {
                        result.add(p.getZip());
                }
        }
} // searchAll()
```

Question 3 (10+5 = 15 points)

11

```
public static ArrayList<String> searchAll(ArrayList<Place> p, String town, String zip) {
        ArrayList<String> result = new ArrayList<String>();
        for (Place pl : p) {
                if (town.equalsIgnoreCase(p.getTown())
                  && !zip.equalsIgnoreCase(p.getState()) {
                        result.add(p.getZip());
                }
        }
} // searchAll()
```

(B) For the above function, what will be its asymptotic time complexity?

O(n), where n = p.size()

Question 4 (5+5=10 points)

12

Consider the statements shown below:

```
int N = …;
ArrayList<Integer> L = new ArrayList<Integer>();

for (int i=1; i <= N; i++) {
   L.add(2*i);
}
```

A. Write commands below that use an <u>iterator</u> to print out all the items in the list to the console window (one item per line):

```
Iterator<Integer> iter = new L.iterator();
while (iter.hasNext()) {
        System.out.println(iter.next());
}
```

Question 4 (5+5=10 points)

13

Consider the statements shown below:

```
int N = …;
ArrayList<Integer> L = new ArrayList<Integer>();

for (int i=1; i <= N; i++) {
   L.add(2*i);
}
```

B. Write commands below that use the enhanced for-loop to print out all the items in the list to the console window (one item per line):

```
for (Integer x : L) {
        System.out.println(x);
}
```

Question 5 (10 points)

14

Implement the add() function, as defined by the List<E> interface, in a singly-linked list that has both a head and a tail in its definition. Assume any helper functions in MyList<E> class that were defined in your text/class are available, including functions for the Node<E> class. Draw before/after diagrams of the data structure.:

```
public class MyList<E> implements List<E> {
   private Node<E> head, tail;
   private int size;
  …
```

Question 5 (10 points)

15

Implement the add() function, as defined by the List<E> interface, in a singly-linked list that has both a head and a tail in its definition. Assume any helper functions in MyList<E> class that were defined in your text/class are available, including functions for the Node<E> class. Draw before/after diagrams of the data structure.:

```
public class MyList<E> implements List<E> {
  private Node<E> head, tail;
  private int size;
 …

 public boolean add(E item) {
  …
 } // add()
```

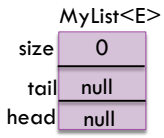
Question 5 (10 points)

16

Implement the add() function, as defined by the List<E> interface, in a singly-linked list that has both a head and a tail in its definition. Assume any helper functions in MyList<E> class that were defined in your text/class are available, including functions for the Node<E> class. Draw before/after diagrams of the data structure.:

```
public class MyList<E> implements List<E> {
  private Node<E> head, tail;
  private int size;
 …

 public boolean add(E item) {
  …
 } // add()
```



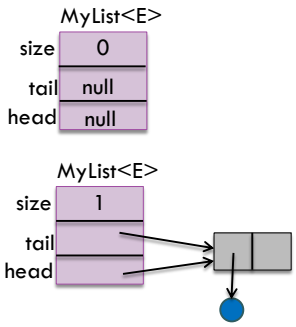
Question 5 (10 points)

17

Implement the add() function, as defined by the List<E> interface, in a singly-linked list that has both a head and a tail in its definition. Assume any helper functions in MyList<E> class that were defined in your text/class are available, including functions for the Node<E> class. Draw before/after diagrams of the data structure.:

```
public class MyList<E> implements List<E> {
  private Node<E> head, tail;
  private int size;
 …

 public boolean add(E item) {
  …
 } // add()
```

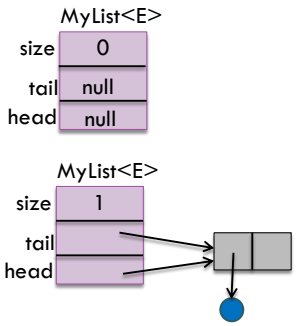


Question 5 (10 points)

18

Implement the add() function, as defined by the List<E> interface, in a singly-linked list that has both a head and a tail in its definition. Assume any helper functions in MyList<E> class that were defined in your text/class are available, including functions for the Node<E> class. Draw before/after diagrams of the data structure.:

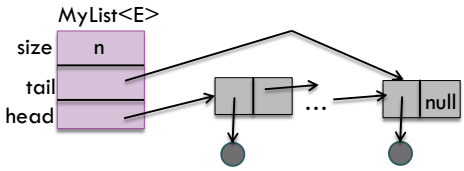
```
public class MyList<E> implements List<E> {
  private Node<E> head, tail;
  private int size;
 …
 public boolean add(E item) {
    if (size==0) {
        head = tail = newNode(item);
    }
    else {
        …
    }
    size++;
 } // add()
```



Question 5 (10 points)

19

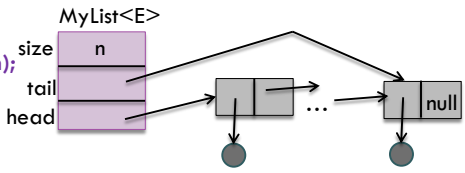
```
public boolean add(E item) {
    if (size==0) {
        head = tail = newNode(item);
    }
    else {
        …
    }
    size++;
    return True;
} // add()
```

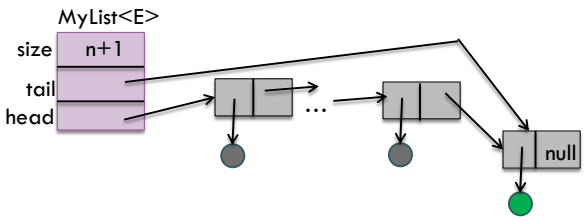


Question 5 (10 points)

20

```
public boolean add(E item) {
    if (size==0) {
        head = tail = newNode(item);
    }
    else {
        Node<E> n=newNode(item);
        tail.next = n;
        tail = tail.next;
    }
    size++;
    return True;
} // add()
```





Question 6 (10 points)

21

- Show the result of evaluating the postfix expressions:

- **11 9 -** =____2_____
- **11 9 – 4 *** =____8_____
- **4 3 + 2 – 6 2 + *** =____40____
- **3 4 7 * 2 / +** =____17_____
- **4 2 2 + *** =____16_____

Question 7 (5+2+3 = 10 points)

22

Consider the commands below:

```
LinkedList<String> potusList = new LinkedList<String>();

String[] presidents = {"Reagan", "Bush41", "Clinton",
                       "Bush43", "Obama", "Trump"};

for (president : presidents) {
   potusList.add(president);
}
```

A. What will be the result of:

potusList.size() ___6_____

potusList.empty() ___False__

potusList.get(1) _"Bush41"_

potusList.get(7) IndexOutOfBoundsException

potusList.contains("Lincoln") __False___

Question 7 (5+2+3 = 10 points)

23

Consider the commands below:

```
LinkedList<String> potusList = new LinkedList<String>();

String[] presidents = {"Reagan", "Bush41", "Clinton",
                       "Bush43", "Obama", "Trump"};

for (president : presidents) {
   potusList.add(president);
}
```

B. What will be output when the following code is executed:

```
for (int i=0; i < potusList.size(); i++) {
   System.out.println(potusList.get(i));
}
```

Reagan
Bush41
Clinton
Bush43
Obama
Trump

Question 7 (5+2+3 = 10 points)

24

Consider the commands below:

```
LinkedList<String> potusList = new LinkedList<String>();

String[] presidents = {"Reagan", "Bush41", "Clinton",
                       "Bush43", "Obama", "Trump"};

for (president : presidents) {
   potusList.add(president);
}

for (int i=0; i < potusList.size(); i++) {
   System.out.println(potusList.get(i));
}
```

C. Rewrite the commands above, using the enhanced-for loop. Other than its appearance, is there a difference between these two ways of writing this loop?

```
for (String p : potusList)
    System.out.println(p);
```

It is much more efficient (O(n) vs O(n^2))

Question 8 (10 points)

25

The summation of the following series:

$$\sum_{i=0}^{n} i = 0 + 1 + \cdots + n$$

can be recursively defined as:

$$sum(n) = \begin{cases} 0, & if\ n = 0 \\ n + sum(n-1), & o/w \end{cases}$$

Write a complete recursive Java function sum(n) that carries out this computation as described above.

```
public static int sum(int sum) {
   if (n == 0)
      return 0;
   else
      return n + sum(n-1);
} // sum()
```

Question 9 (5+5=10 points)

26

A. What is a static variable in a class? Describe one use for it.

Static variables are shared variables, aka class variables. They are different from instance variables since, there is only one "copy" of a static variable that all instance of the class share. They are used in any situation where all instances need to share the same value.

B. What is printed by the command below:

```
int N = (int) (1 + Math.random(100));
System.out.println("I have " + N + (N == 1 ? "item" : "items") + ".");
```

Depends on the value of N.

I have 1 item. When N=1
I have X items When N=X, X>1

27